Mobile Broadband: An Analysis and Improvement of Client-Side Connections to Mobile-Operators in KDE[1]

Peder Osevoll Midthjell
Midthjell
NO-6776 Kjoelsdalen
+0047 41 10 63 95
osevoll@gmail.com

Sveinung Dalatun
Porsmyrvegen 143
NO-5610 Oeystese
+0047 41 45 28 66
sdalatun@gmail.com

Anders Sandven
Vavoll 63
NO-5610 Oeystese
+0047 99 54 30 61
anders.sandven@gmail.com

ABSTRACT

This is a summary of our bachelor's thesis concerning mobile broadband in KDE. The project is divided in two parts. The initial- and final task descriptions do not match, however, they do have the same aim; improve support for mobile broadband in KDE. First the initial task is described, followed by the change of focus. Our solution is presented in the end. We kept the test results from the initial task description as a part of the project, the rest is part of the new task definition.

INTRODUCTION

This project was offered to us by Nokia Qt Software with Knut Yrvin as our external contact person. We were given a lot of freedom when it came to defining the task as most the problem description (i) given by Qt Software ranged widely, making it abstract. We narrowed the problem description down to fit e.g. our time schedule, which resulted in our Initial Task Definition.

Problem Description

Connecting a mobile telephone or a laptop with free software to an operator that offers mobile broadband can be plunder and sticking. Settings and other cryptic information must be entered on the mobile telephone or the PC to establish contact. The task is to simplify access to the mobile operators by fresh install, both on a PC and a mobile telephone.

Today, smart mobile telephones have the same capacity as a PC had 10 years ago. Using free software and Qt, one will take full control over the mobile telephone. The project will program cross-platform software which facilitates access to mobile broadband. One will continue to build on work already done by others as free software. The project will also make an evaluation of what is the easiest way to establish a stable mobile Internet connection considering price, bandwidth and functionality.

Initial Task Definition

We will examine what problem(s) exist in relation to current hard- and software solutions for mobile broadband in relation to KDE. Initially there will be a test phase where we test the dongles, trying to locate the problem(s). The results of the testing are elaborated in an analysis phase where we decide how to solve the problem(s), hopefully we will be able to do this in terms of code. It is also desirable that we investigate the various service providers in Norway in relation to price, bandwidth and functionality.

Due to the tight project time schedule, we could not investigate all aspects of network management in GNU/Linux. This management of network is, according to KDE, ordered in a structure of four layers (See Figure 1), called “The Stack” [10]. We decided to focus on the two uppermost layers; Front-end and NetworkManager [8] (NM) that is.

METHODOLOGY

Before we began the test phase we made a plan for our solution, how to integrate with existing solutions and different design options according to the problem(s) we found. We made a simple sketch of how a possible architecture solution could look like (See Figure 2). There are two lower layer components taking care of, respectively, the detection and solving of a problem. When a problem is detected it would be handled by a logic component that manages the interaction between the components. User would at any time be able to view and choose actions on what to be done via the GUI component.

The design options was:

1. Create a new front-end application if the problem(s) was found in the front-end layer.
2. Change/improve the NM layer and use existing front-end solution if the problem(s) was found in the NM layer.

[1] Bachelor's thesis at Bergen University College, Faculty of Engineering, Department of Computer Engineering, Specialization of System and Software Engineering. Spring 2009.

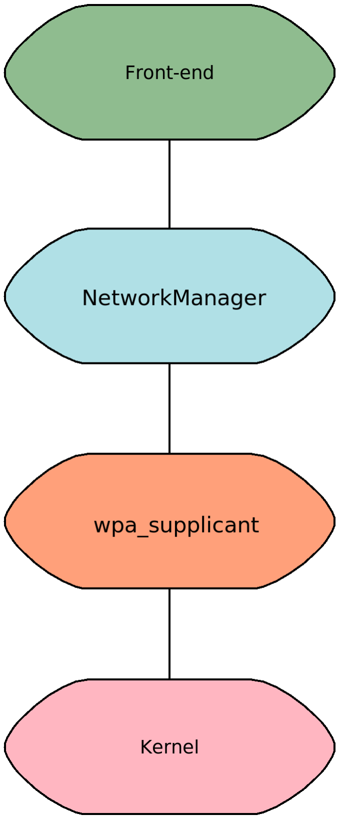


Figure 1: The Stack

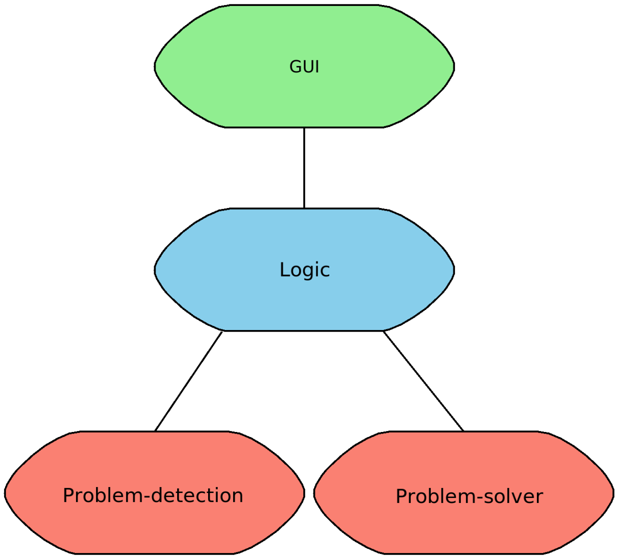


Figure 2: Possible architecture solution

For our solution we made certain requirements; use Qt version 4.5, use conventions from KDE programming, plasmoids instead of standalone programs and of course that it should run under KDE.

In the test phase we tested the dongles from ICE.net [3], Telenor [12] and Netcom [6] in three different application; Network Management Plasmoid [7] (NMP), wvdial [13] and nm-applet [9]. In addition we had a look at the initial impression; what happened the first time we plugged the dongles into the computer, and whether the dongles came with a native GNU/Linux client or not. The dongles was tested individually, meaning no more than one dongle was connected at the same time. The testing was performed with two netbooks; Acer One and Asus Eee, both with Kubuntu as operating system. To gain better understanding of the results during the test phase we studied the code behind NMP and NM. This study made us more capable of analyzing the results in the analysis phase.

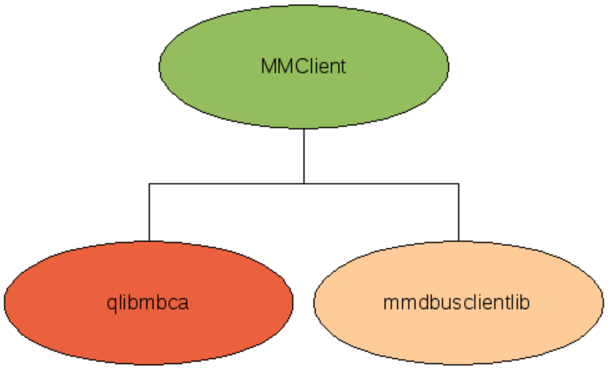
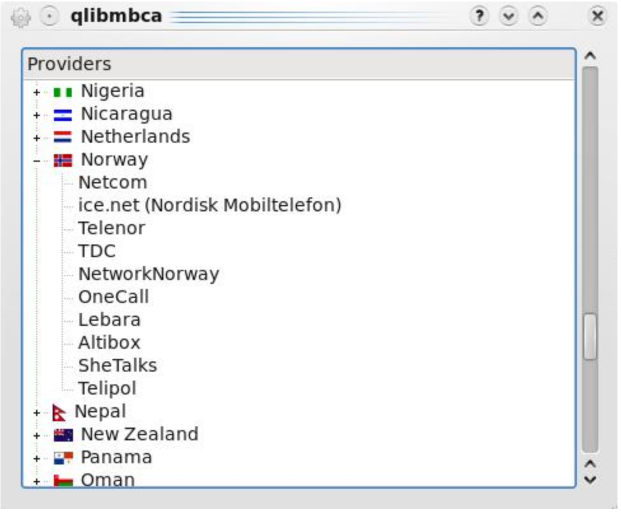


Figure 3: Implemented architecture

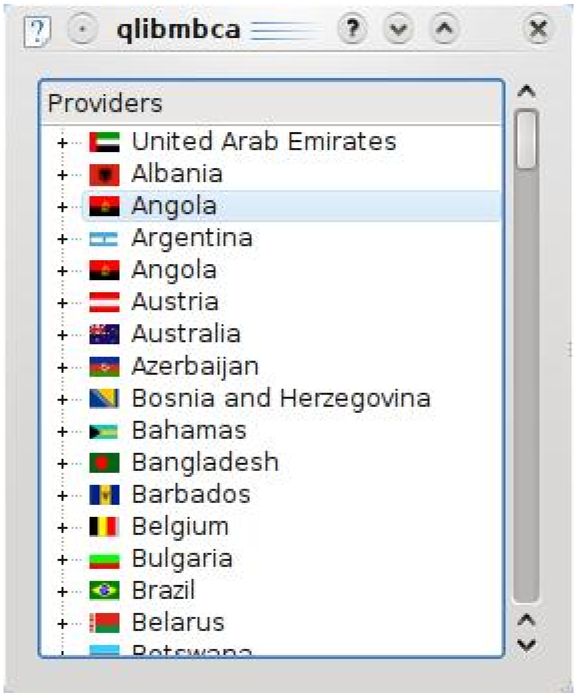
The results of the analysis phase did not turn out the way we firstly had assumed. The support for mobile broadband in KDE (and GNU/Linux in general) is very varying. A lot of dongles need some tweak to make them being recognized as a modem and thus work (more or less). We discovered that the support for mobile broadband is limited in NM and to some extent incomplete in NMP.

The project had a change of path when we realized how big the problem with mobile broadband in KDE really was, and that none of our design options was proper for further work. Having spoken with Will Stephenson, we decided to attend a rejected Google Summer of Code (GsoC) proposal (rejected as in no one chose it). Put in brief the motivation and goal of this proposal are; "Netbooks in these days are common place. Most of them have mobile broadband support (cellular networking) and each day more new people are using them, so KDE should have an easy UI to connect with them, to make the use of this new kind of connections as easy as wifi networks are now. This project is exactly about that, creating an UI to ease the user the usage of those kind of networks." We immediately made a development plan for this proposal, designed- and implemented our solution of it. The architecture of our implemented is illustrated in Figure 3. It consists of 3 modules; qlibmbca, mmdbusclientlib and MMClient. (elaborated in Results). The ultimate goal is to integrate our solution in NMP.



+ Nigeria
+ Nicaragua
+ Netherlands
- Norway
Netcom
ice.net (Nordisk Mobiltelefon)
Telenor
TDC
NetworkNorway
OneCall
Lebara
Altibox
SheTalks
Telipol
+ Nepal
+ New Zealand
+ Panama
+ Oman

Figure 4: qlibmbca (tree-view open)



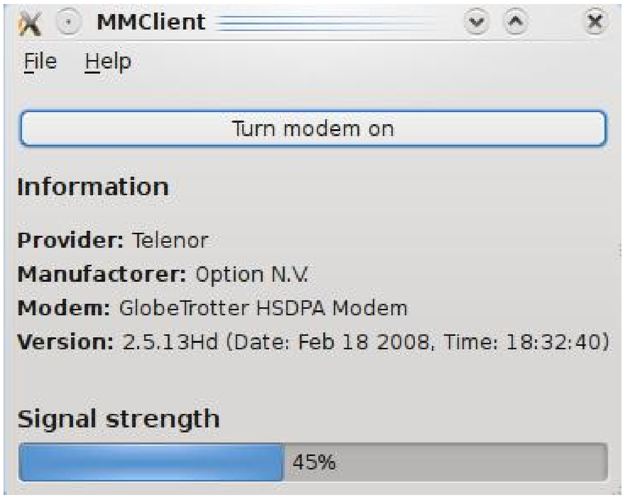
+ United Arab Emirates
+ Albania
+ Angola
+ Argentina
+ Angola
+ Austria
+ Australia
+ Azerbaijan
+ Bosnia and Herzegovina
+ Bahamas
+ Bangladesh
+ Barbados
+ Belgium
+ Bulgaria
+ Brazil
+ Belarus
+ Botswana

Figure 5: qlibmbca (tree-view closed)

During the development phase we used various developer tools. KDevelop [4] was chosen as IDE, ArgoUML [1] for modeling, and Subversion [11] for version control. Due to the project's change of focus we decided to use some of the principles from agile development; good co-operation and dialog within the group, and rapid development at the expense of full documentation.

RESULTS

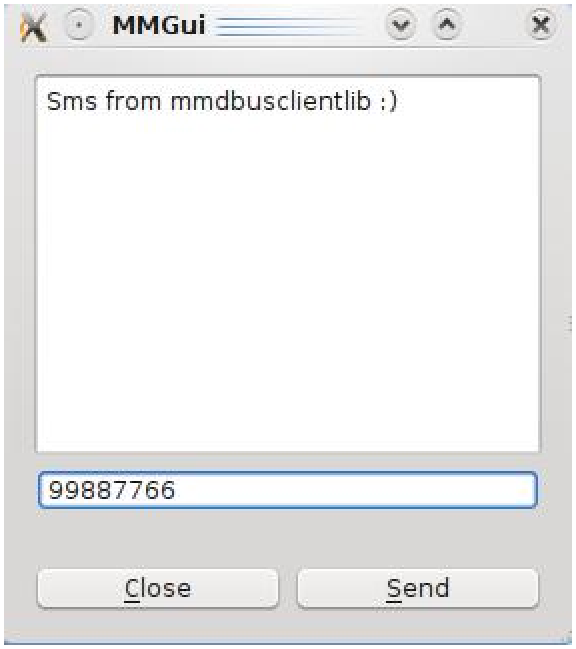
We managed to implement the three modules mentioned in the end of The Methodology section. This section describes what they do and how they turned out in the end.



Provider: Telenor
Manufactorer: Option N.V.
Modem: GlobeTrotter HSDPA Modem
Version: 2.5.13Hd (Date: Feb 18 2008, Time: 18:32:40)

45%

Figure 6: MMClient



Sms from mmdbusclientlib :)

99887766

Figure 7: Send SMS dialog

qlibmbca

qlibmbca is a re-implementation of GNOME's libmbca library, using C++/Qt. This module is an XML-parser, retrieving information from a XML-database file containing service-provider specific information. The library uses a set of dataclasses for the containment of service-provider related data. The XML-parser returns objects of these when it retrieves the service-providers from the database. This results in a list of countries and their corresponding service-providers. In addition this module provides a widget where one can select country and service-provider from a tree-view (See Figures 4 and 5).

mmdbusclientlib

mmdbusclientlib is a library used to communicate with

ModemManager [5] (MM). The purpose of this module is to abstract/hide all D-Bus [2] logic for the GUI developers. Using this library one can easily obtain information such as; signal strength and modem information. An additional feature is to send SMS. Other features that are under development includes contact list and the possibility to receive (and read) SMS.

MMClient

MMClient (See Figure 6) is the name of our proof-of-concept client. It is a simple front-end for the two other modules, used only for testing and demonstration purposes. It was never designed, only implemented alongside the two other modules as new features was completed there. Figure 7 shows a simple dialog box used for sending SMS.

SUMMARY

Mobile broadband is a growth market in which more and more players come along. Using mobile broadband in KDE is plunder and sticking, it is just not to plug the dongle into the computer and start surfing on the Internet. With the power of ModemManagers we are about to fixed this problem. We have implemented the part handling mobile broadband for KDE so that no matter how experienced any end-user may be, he/she should be able to use it with ease. There are still a few features missing (some under development, others to be developed) but the base functionality is there. Hopefully, given a little more time, our solution will mature and be integrated in the network management plasmoid so that everyone gets the benefit of our work with mobile broadband.

ACKNOWLEDGEMENTS

There have been a lot of contributors to this project. A special thanks to Nokia Qt Software for offering this project to Bergen University College, who again offered it to us. Qt Software for hosting the developer sprint with KDE. A big thanks to KDE e.V. For sponsoring the developer sprint, without their help we would not have got as far as we did. Knut Yrvin (external contact person, Qt Software) for all collaboration and an educational time. Lars Michael Kristensen (internal contact person, Bergen University College) for great guidance and follow-up. Will Stephenson, especially for help with planning, developing and debugging. Alexander Sack for debugging and help getting NetworkManager and ModemManager working on our test hardware. Tambet Ingo for help making the modems being recognized by ModemManager, especially for writing HAL-patches and modem test-scripts. Daniel Hess and Holger Levsen (Debian-Edu developers) for help with wiki- and Alioth accounts and the problems concerning these. The KDE developers Frederik Gladhorn and Dario Freddi for all their help, and also the rest of the KDE community.

REFERENCES

1.ArgoUML. http://argouml.tigris.org/

2.D-Bus. http://www.freedesktop.org/wiki/Software/dbus

3.ICE.net. http://www.ice.no/

4.KDevelop. http://www.kdevelop.org/

5.ModemManager. https://launchpad.net/modemmanager

6.Netcom. http://netcom.no/

7.Network Management. http://techbase.kde.org/Projects/Network_Management

8.NetworkManager. http://www.gnome.org/projects/NetworkManager

9.nm-applet. http://www.gnome.org/projects/NetworkManager

10.The Stack. http://userbase.kde.org/NetworkManagement#The_Stack

11.Subversion. http://subversion.tigris.org/

12.Telenor. http://telenor.no/

13.wvdial. http://alumnit.ca/wiki/index.php?page=WvDial